5-1

作用域是一个标识符在程序正文中有效的区域，讨论的是标识符的有效范围。C++的作用域有函数原型作用域、局部作用域（块作用域）、类作用域和命名空间作用域。

5-2

可见性是标识符的有效范围，表示从内层作用域向外层作用域“看”时能看到什么。

可见性的一般规则：

①标识符要声明在前，引用在后

②在同一作用域中，不能声明同名的标识符

③在没有互相包含关系的不同作用域中声明的同名标识符，互不影响

④若在两个或多个具有包含关系的作用域中声明了同名标识符，则外层标识符在内层不可见

5-5

类的静态数据成员是类的数据成员的一种特性，采用static关键字来声明。

特点：

①对于类的普通数据成员，每一个类的对象都拥有一份存储，就是说每个对象的同名数据成员可以分别存储不同的数值，这也是保证对象拥有自身区别于其他对象的特征的需要，但是静态数据成员，每个类只要一份存储，由所有该类的对象共同维护和使用，也就实现了同一类的不同对象之间的数据共享

②静态数据成员具有静态生存期。且由于静态数据成员不属于任何一个对象，因此可以通过类名对它进行访问，一般的用法是“类名::标识符”

5-6

使用static关键字声明的函数成员是静态的。

特点：

①静态函数成员属于整个类，同一个类的所有对象共同维护，为这些对象所共享

②由于静态成员函数只能访问同一个类的静态数据成员，可以保证不会对该类的其余数据成员造成负面影响

③同一个类只维护一个静态函数成员的拷贝，节约了系统的开销，提高程序的运行效率

5-9

①类B不是类A的友元，因为友元不具有交换性

②类C不是类A的友元，因为友元不具有传递性

③类D不是类B的友元，因为友元不具有继承性

5-10

可以

private:

static int a;

5-16

编译输入的是源文件，输出的是目标文件；连接输入的是目标文件，输出的是可执行文件。

编译器对源代码进行编译，是将以文本形式存在的源代码翻译为机器语言形式的目标文件

的过程。连接是将各个编译单元的目标文件和运行库中被调用的单元加以合并后生成的可执行文件的过程。

（1）编译时报错，因为函数参数不匹配

（2）连接时报错，因为函数为定义

（3）不报错

（4）连接时报错，因为函数被重复定义

6-6

\*是指针运算符，也称解析，表示获取指针所指向的变量的值，是一个一元操作符

&是取地址运算符，用来得到一个对象的地址，也是一个一元运算符

6-7

指针是一种数据类型，具有指针类型的变量称为指针变量

指针中存储的地址是所指向对象的地址，而这个地址中的值是所指向对象的值

6-9

“\0”

6-11

引用是一个别名，不能为NULL值，不能被重新分配；指针是一个存放地址的变量。当需要

对变量重新赋以另外的地址或赋值为NULL时只能用指针。

6-15

const int \*p1声明了一个指向整型常量的指针p1，不能通过改变指针p1改变它所指向的整型值，但指针p1本身可以改变，可以指向另外的对象

int \*const p2声明了一个指针型常量，用于存放整型变量的地址，但指针本身的值不能改变，即一旦初始化后，就不能被重新赋值了

6-18

#include <iostream>

using namespace std;

int fnl();

int main()

{

Int \*a=fnl();

Cout<<”the value of a is”<<a;

delete a;

return 0;

}

int fnl()

{

int \*p=new int(5);

return \*p;

}

6-19

long(\*p\_fnl)(int);

long(A::\*p\_fn2)(int);